สนุกกับจอภาพกราฟิกด้วยปาสคาล

อาจหาญ สัตยารักษ์

เทคนิคพิเศษในการแสดงภาพกราฟิกแบบต่าง ๆ โดยใช้เทอร์โบปาสคาลในการเขียนโปรแกรม

เมื่อเราสร้างภาพกราฟิกเก็บไว้ในแผ่นบันทึก ต่อมาเมื่อเราต้องการนำภาพนั้น ๆ มาแสดงบนจอภาพก็ได้ด้วยการเรียกภาพนั้น ๆ มาแสดงตามคำสั่ง PutImage (0,0, Pic, 0) ถ้าเรากำหนดให้ภาพที่เรียกขึ้นมานั้นเป็นตัวแปรชื่อ Pic เป็นต้น ซึ่งจะเป็นการแสดงภาพทีเดียวทั้งจอภาพเมื่อภาพมีขนาดใหญ่สุด

หากเราต้องการให้ภาพที่เราเรียกมานั้นมาซ้อนทับกับภาพเดิมในลักษณะที่แปลก ๆ ออกไป ก็ต้องเขียนชุดคำสั่งช่วย เพื่อให้ทำงานตามต้องการ

บทความนี้มีจุดประสงค์หลักดังกล่าวนี้อย่างครบถ้วน เพียงแต่ผู้เขียนเสนอแนวทางมาให้ 7 แบบก่อน แบบอื่น ๆ ที่นอกเหนือจากนี้ขอทิ้งไว้ให้ผู้อ่านลองคิดดูเองบ้าง เพราะเมื่อทราบหลักการบางแบบก็คงจะสามารถไปพัฒนาให้เป็นรูปแบบอื่น ๆ ได้

คำสั่งปาสคาล (จะเป็น 4.0 หรือ 5.0 ก็ได้) ที่จะใช้คือ

1. SetVisualPage เพื่อกำหนดหน้าที่กราฟิกที่จะใช้แสดงภาพ (มีค่า 0 หรือ 1)

2. SetActivePage เพื่อกำหนดหน้ากราฟิกที่จะใช้งาน (มีค่า 0 หรือ 1)

3. ImageSize เพื่อหาขนาดของภาพเพื่อจะนำไปกำหนดหน่วยความจำให้เก็บภาพ

4. GetMem เพื่อจองเนื้อที่ในหน่วยความจำตามขนาดที่คำนวณได้ โดยจะจองไว้เพียงครั้งแรกครั้งเดียวเท่านั้น

5. GetImage เพื่อนำภาพที่กำหนดเก็บลงในหน่วยความจำ และชื่อภาพที่กำหนด

6. PutImage เพื่อนำภาพที่กำหนดมาแสดง ณ จุด (มุมบนซ้าย) ที่กำหนด

กำหนดรูปแบบเบื้องต้น

ผู้เขียนขอให้แนวคิดง่าย ๆ ในช่วงนี้ก่อนว่าการแสดงภาพนั้นเราจะทำดังนี้

1. กำหนดหน้าที่กราฟิกเริ่มต้นที่หน้า 0 โดยคำสั่ง SetVisualPage(0);

2. กำหนดใช้กราฟิกหน้า 1 คือ setActivePage(1); แล้วเรียกภาพมาแสดงไว้ที่กราฟิกหน้านี้ก่อน เพื่อป้องกันการแสดงภาพที่กวาดจากบนลงล่างปรากฏบนจอภาพด้วยคำสั่ง PutImage

3. เรียกกราฟิกหน้า 1 มาแสดงภาพบนจอด้วยคำสั่ง SetVisualPage (1); ซึ่งจะเห็นภาพทั้งจอได้ทันที

4. ช่วงนี้อาจแอบนำภาพไปแสดงที่กราฟิกหน้า 0 อีกก็ได้

5. เมื่อต้องการแสดงภาพอื่น ๆ เข้ามาแทนที่ ก็กำหนดให้ภาพนั้นเข้ามาอยู่ในกราฟิกอีกหน้าหนึ่งก่อน เช่นถ้าใช้กราฟิกหน้า 0 แสดงภาพปรากฏบนจอก็เรียกภาพใหม่เข้ามาที่กราฟิกหน้า 1

6. หาทางนำภาพที่อยู่กราฟิกอีกหน้าหนึ่งนั้นมาแสดงในหน้าที่แสดงภาพ

โดยกำหนดคำสั่งสลับกันไปมา เช่น กำหนด setVisualPage(0); เป็นจอแสดงภาพให้ผู้ดูเห็น ภาพที่สองก็จะอยู่ในกราฟิกหน้า 1 ก็สามารถนำภาพนั้นมาแสดงในหน้า 0 ได้ คือ

ก. setActivePage (1); แล้วเก็บภาพส่วนหนึ่งของภาพที่สองไว้ในตัวแปรที่กำหนด

ข. setActivePage(0); แล้วนำภาพที่เก็บไว้ในตัวแปรตามข้อ ก มาแสดงในจอภาพที่ใช้งานและแสดงภาพ

ค. วนไปทำตามข้อ ก และ ข จนกว่าจะได้ภาพครบถ้วนสมบูรณ์

ง. การแสดงภาพมีการทำงานอย่างไรนั้นขึ้นกับการกำหนดขอบเขตของภาพในข้อ ก และการแสดงภาพในข้อ ข ประกอบกัน

ก่อนอื่นก็ขอกำหนดตัวแปรและชุดคำสั่งพื้นฐานอื่นไว้ก่อน ดังในรูปที่ 1

ตัวแปรที่กำหนดไว้มีดังนี้

1. p1,p2,p3,p4 เพื่อจะนำไปเป็นเก็บภาพ 4 ภาพ (เป็นเพียงพอยเตอร์ชี้ไปตำแหน่งเก็บภาพ)

2. Size ตัวแปรเพื่อกำหนดขนาดของภาพซึ่งจะกำหนดโดยใช้คำสั่ง Size:= ImageSize

3. Pic, Pic1 เป็นตัวแปรเพื่อเก็บภาพอีกแนวทางหนึ่งการกำหนดแบบนี้แตกต่างจากข้อ 1 ในแง่ข้อหนึ่งนั้นเป็นเพียงพอยเตอร์ชี้ไปยังตำแหน่งของภาพเท่านั้น ส่วนตัวแปร Pic,Pic1 นี้เป็นกลุ่มของภาพจริง ๆ จึงสามารถใช้เก็บลงแผ่นบันทึกหรือเรียกจากแผ่นบันทึกมาสู่หน่วยความจำ

4. PicFile กำหนดชนิดของไฟล์ที่จะเรียกจากแผ่นบันทึก

ชุดคำสั่งที่กำหนดไว้คือ

1. OpenGraph; เพื่อเรียกกราฟิกของจอโมโนโครม

2. ClsScreen เพื่อใช้ลบจอภาพตามขอบเขตที่กำหนดต่อท้าย

```
{=============== DISPLAY.PAS ================}

USES crt,graph;
  type  PicBuff=array[0..31324] of byte;
  var
      p1,p2,p3,p4:pointer;
      Size:word; Pic,Pic1:PicBuff;
       PicFile:file of PicBuff; ch:char;

PROCEDURE OpenGraph;
var
   GrDriver,GrMode :integer;
begin
   GrDriver :=7; GrMode:=0;
  InitGraph(GrDriver, GrMode, '');
end;

PROCEDURE ClsScreen(x1,y1,x2,y2:word);
  begin;
   SetFillStyle(0,0); Bar(x1,y1,x2,y2); SetFillStyle(1,1);
  end;

PROCEDURE Loadpic;
begin
    setVisualPage(0); SetActivePage(1);
    assign(PicFile,'B:PICTEST.PIC');
      Reset(PIcFile);
      Read(PicFile,Pic);
      PutImage(0,0,Pic,0);
    CLose(PicFile);
    setVisualPage(1);
End;

PROCEDURE Table;
var k:word;
begin
  setVisualPage(0); SetActivePage(1);
  rectangle(0,0,719,347); k:=10;
    Repeat
      line(k,0,k,347); Inc(k,10);
    until k>710;
     k:=10;
    Repeat
      line(0,k,719,k); Inc(k,10);
    until k>340;
   GetImage(0,0,719,347,Pic1);
end;
```

รูปที่ 1 โปรแกรมเริ่มต้นเพื่อกำหนดค่า เข้ากราฟิก อ่านภาพ และวาดภาพ

รูปที่ 2 ภาพข้อความที่ถูกเก็บในแผ่นด้วยชื่อ TESTPIC.PIC

3. LoadPic เพื่อเรียกภาพกราฟิกเข้ามาในหน่วยความจำ โดยจะเรียกเข้ามาที่กราฟิกหน้า 1 แล้วแสดงผลให้ปรากฏในกราฟิกหน้าหนึ่งเช่นกัน (ภาพที่เรียกเข้ามาคือภาพชื่อ PICTEST.PIC สมมติเป็นภาพดังรูปที่ 2) หากท่านเรียกจากไดร์ฟ A ก็ต้องกำหนดเป็น A:PICTEST.PIC ด้วย ภาพที่ถูกเรียกเข้ามาจะอยู่ในตัวแปรที่ชื่อ PIC ตามที่ปรากฏในคำสั่ง PutImage (0,0,Pic,0);

4. Table เพื่อทำการตีตารางให้ปรากฏบนจอภาพ ในที่นี้ผู้เขียนเอาไปแสดงไว้ล่วงหน้าก่อนที่จะแสดงภาพในแต่ละแบบ เพื่อให้มองเห็นการซ้อนทับของภาพ 2 ภาพเท่านั้น ผู้อ่านอาจจะวาดเองหรือเรียกภาพอื่นมาจากแผ่นบันทึกก็ได้

นี่คือโปรแกรมส่วนต้นที่ได้กำหนดไว้ (เป็นส่วนหนึ่งในไฟล์ Display.pas ในซอฟต์เมล์)

การแสดงภาพแบบกะพริบ

การแสดงแบบแรกที่ต้องการให้ศึกษาก็คือ การแสดงภาพแบบกะพริบ ซึ่งเป็นแบบง่ายที่สุด เช่นสมมติว่าผู้เขียนต้องการให้ภาพในรูปที่ 2 นั้นกะพริบที่ข้อความ "ยินดีต้อนรับทุกท่าน" ก็จะทำได้โดยไม่ยาก

สมมติตอนนี้กำลังแสดงภาพบนจอด้วยกราฟิกหน้า 1 ก็กำหนดการทำงานของชุดคำสั่งได้ดังนี้

1. กำหนดกราฟิกหน้า 0 เพื่อเป็นหน้าใช้งาน (ยังไม่ใช่หน้ากราฟิกที่แสดงภาพนะ) ด้วยคำสั่ง setActivePage(0);

2. นำภาพ Pic มาแสดงอีกครั้งที่กราฟิกหน้า 0 นี้

3. หาขอบเขตของภาพในบริเวณที่ต้องการให้กะพริบ (ในที่นี้คือช่วง 0, 90, 400, 147) ลบภาพส่วนนี้ทิ้งไป ในตอนนี้ภาพที่อยู่ในกราฟิกทั้งสองหน้า (0 และ 1) จึงมีลักษณะเหมือนกับยกเว้นส่วนที่ลบทิ้งเท่านั้นที่กราฟิกหน้า 1 มีข้อความว่า "ยินดีต้อนรับทุกท่าน" ส่วนกราฟิกหน้า 0 จะไม่มีข้อความนี้

4. ตอนนี้เราพร้อมจะแสดงการกะพริบของข้อความว่า ยินดีต้อนรับทุกท่าน โดยการวนรอบคำสั่ง Repeat พร้อมกับแสดงกราฟิก 2 ภาพ (2 หน้า) สลับกันไปเรื่อย ๆ จนกว่าจะมีการกดแป้น ซึ่งจะมีผลทำให้ได้ภาพกะพริบของข้อความที่กำหนดไว้ดังกล่าว

5. การกะพริบนั้นจะช้าหรือไวนั้น ขึ้นอยู่กับค่า delay ที่กำหนดไว้ในชุดคำสั่งหลังการแสดงภาพของกราฟิกแต่ละหน้านั่นเอง

ชุดคำสั่ง DispPic_I ในรูปที่ 3 คือการทำงานในเรื่องนี้ ขอให้ศึกษาดู

```
PROCEDURE DispPic_I;
   begin
     ch:=#0;
     setactivepage(0);
     PutImage(0,0,Pic,0);
     ClsScreen(0,90,400,147);
       Repeat
         setVisualPage(0); Delay(500);
         setVisualPage(1); Delay(500);
      if KeyPressed then ch:=readkey;
    Until Ch<>#0;
   End;
```

รูปที่ 3 ชุดคำสั่งการแสดงภาพข้อความแบบกะพริบ

การทำภาพเคลื่อนไหวหรืออักษรวิ่ง

หากการทำภาพกะพริบยังไม่เป็นที่ถูกใจ เราก็อาจทำเป็นภาพเคลื่อนไหวหรืออักษรวิ่งก็ได้ เช่นเราอยากให้ข้อความว่า "ยินดีต้อนรับทุกท่าน" นั้นเปลี่ยนมาเป็นอักษรวิ่งแทนการกะพริบเราก็ทำได้ดังนี้

ขณะนี้กราฟิกที่แสดงภาพให้เห็นบนจอภาพคือกราฟิกหน้า 1 (ตามคำสั่ง setVisualPage(1))

1. กำหนดตัวแปรที่ใช้เก็บภาพ 2 ภาพ ในที่นี้ใช้ตัวแปร p1 และ p2

2. กำหนดรูปแบบการวิ่งว่าจะให้วิ่งแบบไหน เช่น วิ่งจากซ้ายไปขวา จากขวาไปซ้าย หรือบนลงล่าง ล่างขึ้นบน เป็นต้น ในที่นี้ผู้เขียนจะแสดงการวิ่งของข้อความจากขวาไปซ้าย และจะวนรอบไปเรื่อย ๆ จนกว่าจะมีการกดแป้น ซึ่งจะเป็นการวนรอบเฉพาะส่วนที่อยู่ในขอบเขต 0, 90, 719, 147 เท่านั้น

3. กำหนดขอบเขตของภาพออกเป็น 2 ส่วน ดังนี้

p2^	ส่วนของ p1^

ให้ส่วนของ p2 มีขนาดเล็กอยู่ทางซ้ายสุด (0, 90, 19, 147) และส่วนของ p1 คือส่วนที่เหลือถัดมาคือ (20, 90, 719, 147)

4. แอบนำภาพของ p2, p1 มาแสดงใหม่ที่กราฟิกหน้า 0 โดยสลับที่กันดังนี้

แสดงภาพ p1^ ตรงนี้	p2^

เมื่อได้สลับที่ของภาพแล้วก็เรียกให้กราฟิกหน้า 0 เป็นหน้าที่แสดงภาพพร้อมกับไปจำภาพ p1, p2 นี้ใหม่อีกครั้งแบบเดียวกับข้อ 3

5. ทำแบบเดียวกับข้อ 4 แต่คราวนี้เป็นการแอบไปทำภาพที่กราฟิกหน้า 1 (เพราะผู้ดูกำลังดูกราฟิกหน้า 0 อยู่)

6. โดยการวนรอบตามข้อ 3,4,5 ไปเรื่อย ๆ อย่างนี้ เราจะมองเห็นภาพข้อความที่กำหนดวิ่งวนจากขวาไปซ้ายเรื่อย ๆ จนกว่าจะมีการกดแป้นใดแป้นหนึ่ง

ชุดคำสั่งชื่อ DispPic_II ในรูปที่ 4 คือส่วนแสดงผลในลักษณะนี้

สองแบบที่ผ่านมาเป็นการแสดงของภาพ 1 ภาพ คราวนี้สมมติเรามีภาพอยู่ 2 ภาพ โดยภาพที่หนึ่งถูกแสดงแสดงปรากฏอยู่บนจอภาพแล้ว และจะนำภาพที่สองเข้ามาแทนที่ภาพแรกนั้น เราจะมีวิธีทำอย่างไรบ้าง

สมมติว่าภาพตามชุดคำสั่ง Table เป็นภาพที่หนึ่ง และภาพตามรูปที่ 2 นั้นเป็นภาพที่จะเรียกเข้ามาแทนภาพตารางดังกล่าว เราจะมาเขียนชุดคำสั่งการแสดงภาพต่อไป

ในที่นี้ผู้เขียนขอเสนอรูปแบบการแทนที่ของภาพที่สองสัก 5 วิธี ดังต่อไปนี้

การกวาดภาพจากซ้ายไปขวา

การแทนที่แบบนี้จะเป็นการนำภาพที่สองมาแสดงเริ่มต้นจากด้านซ้ายมือแล้วขยับไปเรื่อย ๆ ทางด้านขวามือ จนกว่าจะหมดภาพที่สอง หรืออีกนับหนึ่งก็คือภาพที่หนึ่งจะหายไปเรื่อยจากซ้ายไปขวา

คราวนี้จะใช้ตัวแปรเก็บภาพเพียงตัวแปรเดียวคือตัวแปร p1 เท่านั้น

ขอให้ดูชุดคำสั่ง DispPic_III ในรูปที่ 5 ประกอบ

1. กำหนดค่า xx:=0; ซึ่งหมายถึงด้านซ้ายสุด และ Xpix:=5 ซึ่งหมายถึงการกำหนดความกว้างของแถบภาพ p1 ที่จะนำมาแสดงทีละภาพนั่นเอง (ค่า Xpix ถ้ากำหนดค่ามากขึ้นก็จะได้แถบภาพที่เคลื่อนย้ายกว้างขึ้นตาม)

2. กำหนดกราฟิกหน้า 0 นำภาพที่หนึ่ง (Pic1) ไปเก็บไว้ แล้วให้แสดงภาพนี้บนจอภาพ

3. กำหนดกราฟิกหน้า 1 นำภาพที่สอง (Pic) มาเก็บไว้

4. แอบไปเก็บภาพที่กราฟิกหน้า 1 (ภาพของ Pic) มีขนาดความกว้างเป็น xx,0 ,xx+Xpix,347 มาไว้ในตัวแปร p1^

5. นำภาพ p1^ นั้นมาแสดงที่กราฟิกหน้า 0 ในตำแหน่งเดียวกันคือ xx,0 เป็นมุมบนซ้าย

6. กำหนดค่า xx เพิ่มขึ้นเท่ากับค่า Xpix+1 ด้วยคำสั่ง Inc(xx,Xpix+1); เพื่อจะขยับไปแอบเก็บภาพที่สองจากกราฟิกหน้า 1 ในแถบถัดไปมาแสดงต่อจากภาพเดิมในกราฟิกหน้า 0 ที่ได้แสดงไปแล้ว

7. วนรอบอย่างนี้ไปเรื่อย ๆ จนกว่าจะได้ค่า xx ไปสุดจอภาพด้านขวามือ ก็เป็นอันจบการทำงาน

อย่าลืมว่าการดูบนจอภาพเท่านั้นที่ผู้อ่านจะเห็นการทำงานตามโปรแกรมได้ จบชุดคำสั่งนี้ภาพที่ปรากฏบนจอภาพจะเป็นภาพจากรูปที่สอง(pic) และต่อจากนี้เป็นต้นไปเราจะกำหนดหน้ากราฟิกแสดงผลเป็นหน้า 0 ตลอดไป

การกวาดภาพจากล่างขึ้นบน

การแสดงภาพแบบนี้มีแนวคิดแบบเดียวกันกับการกวาดจากซ้ายไปขวา จึงไม่ขออธิบายขอให้ศึกษาเองจากชุดคำสั่ง DispPic_IV ในรูปที่ 5 เช่นกัน ซึ่งคิดว่าคงเข้าใจได้ไม่ยากนัก เพียงแต่ใช้ตัวแปรต่างออกไปเท่านั้น และกำหนดค่า yy ที่ล่างสุด แล้วเลื่อนค่าขึ้นเรื่อย ๆ (ในโปรแกรมคือการลดค่า yy จากคำสั่งที่เขียนว่า Dec(yy, Ypix)นั่นเอง) และหยุดทำงานเมื่อค่า yy<0 จบส่วนนี้ภาพที่ปรากฏบนจอภาพคือภาพตารางหรือภาพ pic1

```
PROCEDURE DispPic_II;
  begin
     Ch:=#0;
     setVisualPage(1); setactivepage(1);
     GetImage(0,90,19,147,P1^); GetImage(20,90,719,147,p2^);
        Repeat
          setactivepage(0);
          PutImage(0,90,P2^,0); PutImage(700,90,P1^,0);
          setVisualPage(0);
          GetImage(0,90,19,147,P1^); GetImage(20,90,719,147,p2^);
          setactivepage(1);
          PutImage(0,90,P2^,0); PutImage(700,90,P1^,0);
          GetImage(0,90,19,147,P1^); GetImage(20,90,719,147,p2^);
          setVisualPage(1);
         if KeyPressed then ch:=readkey;
       Until Ch<>#0;
  End;
```

รูปที่ 4 ชุดคำสั่งการแสดงภาพอักษรวิ่งจากซ้ายไปขวา

การแสดงภาพแบบเปิดม่านและแบบปิดม่าน

ตอนนี้บนจอภาพจะแสดงภาพตาราง (ซึ่งสมมติว่านี้คือม่านหรือฉาก) เราสามารถนำภาพอีกภาพหนึ่งคือภาพตามรูปที่สองมาแสดงต่อไปแบบการเปิดม่านจากตรงกลางออกไปยังส่วนข้าง ๆ พร้อมกับมีภาพข้อความแสดงแทนที่ได้ ทั้งนี้รวมทั้งสามารถเขียนให้เป็นแบบปิดม่านย้อนกลับได้ด้วย

ชุดคำสั่ง DispPic_V คือชุดสำหรับการเปิดม่าน และชุดคำสั่ง DispPic_VI คือการปิดม่าน ดังที่เห็นในรูปที่ 6

หลักการทำงานคล้าย ๆ กัน ต่างกันแค่ตำแหน่งการเคลื่อนย้ายภาพเท่านั้น จึงขออธิบายเพียงชุดคำสั่งเดียวเท่านั้น คือ DispPic_V ดังนี้

1. กำหนดค่า x1 และ x2 ให้เริ่มที่กึ่งกลางจอภาพคือ ประมาณจุดที่ 359 -360 กำหนดค่าความกว้างของภาพที่จะเก็บมาแสดง (xPix:=5)

2. นำภาพที่จะนำมาแสดงแทนที่ภาพเดิม (ในกรณีนี้คือภาพ pic) ไปเก็บไว้ที่กราฟิกหน้า 1

3. เก็บภาพจากกราฟิกหน้าหนึ่งนี้มาสองแถบในตัวแปรชื่อ p1,p2

4. นำภาพที่เก็บมานั้นมาแสดงที่กราฟิกหน้า 0

5. กำหนดค่า x1,x2 ใหม่โดยลดค่า x1 และเพิ่มค่า x2 แล้วกลับไปเก็บภาพมาอีก 2 แถบในตัวแปรเดิม แล้วมาแสดงในกราฟิก 0 ณ จุดเดียวกัน

6. ทำอย่างนี้ไปเรื่อย ๆ จนกว่าจะได้ภาพมาครบถ้วน (ค่า x2>719-Xpix) ก็จบการวนรอบ

การปิดม่านก็ทำคล้ายกันเพียงแต่กำหนดค่า x1 ที่ซ้ายสุด และ x2 ที่ขวาสุด แล้วเก็บภาพมาทีละสองภาพเช่นกัน ขอให้ทำความเข้าใจจากชุดคำสั่งทั้งสองเปรียบเทียบกัน

```
PROCEDURE DispPic_III;
  Var xx,xpix:Integer;
    begin
      xx:=0; Xpix:=5;
      setactivepage(0);  PutImage(0,0,Pic1,0);
      setVisualPage(0);
      setactivepage(1);  PutImage(0,0,Pic,0);
       Repeat
         setactivepage(1);
         GetImage(xx,0,xx+Xpix,347,P1^);
         setactivepage(0);
         PutImage(xx,0,P1^,0);
         Inc(xx,Xpix+1);
       until xx>719-Xpix;
    End;

  PROCEDURE DispPic_IV;
   Var yy,Ypix:Integer;
   begin
         Ypix:=1;  yy:=348;
         setactivepage(1);  PutImage(0,0,Pic1,0);
         setactivepage(0);  setVisualPage(0);
          Repeat
            Dec(yy,Ypix+1);
            setactivepage(1);
            GetImage(0,yy,719,yy+Ypix,P1^);
            setactivepage(0);
            PutImage(0,yy,P1^,0);
          Until yy<0;
   End;
```

รูปที่ 5 ชุดคำสั่งการแสดงภาพแบบซ้ายไปขวาและล่างขึ้นบน

```
PROCEDURE DispPic_V;
 Var x1,x2,xpix:Integer;
  begin
    x2:=360; Xpix:=5;  x1:=359-Xpix;
    setactivepage(1);  PutImage(0,0,Pic,0);
    setactivepage(0);  setVisualPage(0);
     Repeat
       setactivepage(1);
       GetImage(x1,0,x1+Xpix,347,P1^);
       GetImage(x2,0,x2+Xpix,347,P2^);
       setactivepage(0);
       PutImage(x1,0,P1^,0); PutImage(x2,0,P2^,0);
       Inc(x2,Xpix+1);  Dec(x1,Xpix+1);
    Until x2>719-Xpix;
 End;

PROCEDURE DispPic_VI;
 Var x1,x2,xpix:Integer;
  begin
    x1:=0;  Xpix:=5;  x2:=719-Xpix;
    setactivepage(1);   PutImage(0,0,Pic1,0);
    setVisualPage(0);
      Repeat
        setactivepage(1);
        GetImage(x1,0,x1+Xpix,347,P1^);
        GetImage(x2,0,x2+Xpix,347,P2^);
        setactivepage(0);
        PutImage(x1,0,P1^,0); PutImage(x2,0,P2^,0);
        Inc(x1,Xpix+1);  Dec(x2,Xpix+1);
      until x1>360;
  End;
```

รูปที่ 6 ชุดคำสั่งการแสดงภาพแบบเปิดม่าน-ปิดม่าน

```
PROCEDURE DispPic_VII;
  Var x1,x2,y1,y2,pix:Integer;
    begin
     pix:=5; x1:=360; x2:=364+Pix;
     y1:=174; y2:=174;
     setactivepage(1);  PutImage(0,0,Pic,0);
     setactivepage(0);
      Repeat
          setactivepage(1);
          GetImage(x1,y1,x1+Pix,y2+pix,P1^);
          GetImage(x2,y1,x2+Pix,y2+pix,P2^);
          GetImage(x1,y1,x2+Pix,y1+pix,P3^);
          GetImage(x1,y2,x2+Pix,y2+pix,P4^);
          setactivepage(0);
          PutImage(x1,y1,P1^,0); PutImage(x2,y1,P2^,0);
          PutImage(x1,y1,P3^,0); PutImage(x1,y2,P4^,0);
          Dec(x1,pix); INc(x2,pix); Dec(y1,3); Inc(y2,3);
          If y1<0 then y1:=0;
          if y2>347 then y2:=345;
       Until x1<0;
    End;
```

รูปที่ 7 ชุดคำสั่งการแสดงภาพแบบวนตรงกลางออกมาหาขอบ

p1+p3	ส่วนของภาพ P3^	p2+p3
ส่วนของ P1^	บางส่วนของภาพที่ถูกเก็บมาแสดงก่อนหน้านี้แล้ว	ส่วนของ p2^
p1+p4	ส่วนของภาพ p4^	p2+p4

รูปที่ 8 แสดงการเก็บลงในตัวแปร P1^ - P4^

```
{========================== MAIN ==========================}
 begin
  OpenGraph;
     size:=ImageSize(0,0,719,347);
     GetMem(P1,Size);
     size:=ImageSize(0,0,719,50);
     GetMem(P3,Size); GetMem(P4,Size); GetMem(P2,Size);
     Table; LoadPic;
     DispPic_I;
     DispPic_II;
     DispPic_III;
     DispPic_IV;
     DispPic_V;
     DispPic_VI;
     DispPic_VII;
     ch:=readkey;
  CloseGraph;
end.
```

รูปที่ 9 ชุดคำสั่งโปรแกรมหลักเพื่อการทำงานทั้งหมดของการแสดงภาพ

การแสดงภาพวนออกจากตรงกลาง

ตอนนี้ภาพบนจอจะเป็นภาพของตาราง (pic1) อยู่

คราวนี้มาถึงแบบสุดท้ายคือแบบที่ 7 (มีแบบอื่น ๆ อีกมากมายที่จะแสดงได้ ขอให้ผู้อ่านลองคิดดู) เป็นลักษณะการแทนที่ภาพแบบเริ่มที่ตรงกึ่งกลางจอภาพ แล้วขยายขอบเขตโตขึ้นทั้งสี่ด้านจนกระทั่งได้ภาพใหม่เต็มจอภาพ

เราจะนำภาพข้อความกลับมาแสดงที่จอภาพใหม่อีกครั้งหนึ่ง ซึ่งคราวนี้เราต้องใช้ตัวแปรในการเก็บภาพมาแสดงทั้งหมด 4 ตัวแปร คือ p1,p2,p3 และ p4 ซึ่งมีการทำงานเป็นดังนี้

1. กำหนดค่า xt,x2,y1,y2 เริ่มที่ตรงกึ่งกลางภาพ

2. นำภาพ (pic) ไปไว้ที่กราฟิกหน้า 1

3. เก็บภาพจากกราฟิกหน้าหนึ่งมาตามรูปที่ 8 (สมมติว่าเก็บมาได้ส่วนหนึ่งแล้ว)

4. นำภาพทั้งสี่นี้ไปแสดงในกราฟิกหน้า 0 และเปลี่ยนแปลงค่า x1, x2,y1,y2 แล้วกลับไปเก็บภาพในกราฟิกหน้า 1 มาแสดงอีกจนกว่าจะได้ภาพครบถ้วน (เมื่อค่า x1<0) ก็จบการทำงาน

เมื่อนำโปรแกรมในรูปที่ 1,3,4,5, 6,7 และโปรแกรมส่วนเรียกชุดคำสั่งทั้งหมดมาทำงานในรูปที่ 9 มารวมกันนำเก็บลงในแผ่นให้ชื่อว่า display.pas ก็จะได้โปรแกรมการแสดงภาพ 2 ภาพอย่างต่อเนื่องกันไปที่สมบูรณ์พร้อมที่จะทำงานได้

หวังว่าคงมีประโยชน์ต่อผู้รักและหลงใหลในกราฟิกไม่มากก็น้อย